A Partially Binarized Hybrid Neural Network System for Low-Power and Resource Constrained Human Activity Recognition

Antonio De Vita, *Graduate Student Member, IEEE*, Alessandro Russo, *Graduate Student Member, IEEE*, Danilo Pau, *Fellow, IEEE*, Luigi Di Benedetto, *Member, IEEE*, Alfredo Rubino, *Member, IEEE*, and Gian Domenico Licciardo, *Senior Member, IEEE*

***Abstract*— A custom Human Activity Recognition system is presented based on the resource-constrained Hardware (HW) implementation of a new partially binarized Hybrid Neural Network. The system processes data in real-time from a single tri-axial accelerometer, and is able to classify between 5 different human activities with an accuracy of 97.5% when the Output Data Rate of the sensor is set to 25 Hz. The new Hybrid Neural Network (HNN) has binary weights (i.e. constrained to +1 or −1) but uses non-binarized activations for some layers. This, in conjunction with a custom pre-processing module, achieves much higher accuracy than Binarized Neural Network. During pre-processing, the measurements are made independent from the spatial orientation of the sensor by exploiting a reference frame transformation. A prototype has been realized in a Xilinx Artix 7 FPGA, and synthesis results have been obtained with TSMC CMOS 65 nm LP HVT and 90 nm standard cells. Best result shows a power consumption of 6.3 μW and an area occupation of 0.2 mm^2 when real-time operations are set, enabling in this way, the possibility to integrate the entire HW accelerator in the auxiliary circuitry that normally equips inertial Micro Electro-Mechanical Systems (MEMS).**

***Index Terms*— Artificial neural networks, human activity recognition, digital HW design, low power, inertial sensor.**

I. Introduction

ONE of the most interesting features in modern portable and wearable devices is Human Activity Recognition (HAR). It refers to the ability of a system to identify the activities executed by a person, by processing data acquired from a set of sensors, which monitor movements of parts of the human body [1]. Both image and inertial sensors are used to build HAR systems [2], [3], but the evolution of MEMS has largely contributed to make inertial-based HAR systems more cost-efficient than the image-based counterpart [4]. Therefore, HAR systems, based on accelerometers, gyroscopes or heterogeneous Inertial Measurement Units (IMUs), are becoming very popular in a number of handheld, embedded, and wearable devices [5]. They are used in a wide range of applications, ranging from medical to personal, such as Parkinson's disease monitoring, rehabilitation, microsurgical devices, fall detection and fitness [6]–[10]. Recently, the need for increased recognition capabilities paved the way to the exploitation of AI paradigms. They are starting to be deployed in HAR by using Pattern Recognition (PR) models (decision tree, support vector machine, naïve Bayes) [11] or Deep Learning (DL) models [12], [13]. Although DL enables superior recognition capabilities and accuracy than PR [14], the latter is often employed in resource constrained devices for its lower computational complexity [2], [15]. Cloud computing could be used to move away part of the computational load of DL from local devices [16], [17]. However, it is not an optimal solution because of the additional energy budget required for communications, and the performance degradation due to bandwidth limitations. Therefore, keeping the computation close to the sensing element, according to the edge computing paradigm, appears the most viable solution to minimize delay and power consumption [18]. In order to effectively deploy DL in compact HAR systems [19], reduced precision models of Artificial Neural Networks (ANNs) have been proposed in the recent years [20], [21], which quantize weights and activations to lower precisions than the standard Floating-Point (FP) [22]. The extreme case leads to Binarized Neural Networks (BNNs) [23], where both weights and activations are constrained to +1 or −1, and coded by only 1 bit. This significantly reduces the memory requirements, and enables the elimination of multiplications [24]. However, the significant accuracy loss of BNNs is only partially mitigated by new specific training techniques [25], [26] and seriously limits their employment in HAR, where accuracy is a primary requirement.

In order to overcome the above limitations, for the first time in the literature, in this article a new HW accelerator of HAR features is presented. It exhibits area and energy requirements small enough to be compatible with the integration into the auxiliary circuits of the sensors [27]. The system is built

Manuscript received March 14, 2020; revised June 12, 2020 and July 22, 2020; accepted July 22, 2020. This work was supported by the Italian Ministry of Education, University and Research. This article was recommended by Associate Editor D. John. *(Corresponding author: Gian Domenico Licciardo.)*

Antonio De Vita, Alessandro Russo, Luigi Di Benedetto, Alfredo Rubino, and Gian Domenico Licciardo are with the Department of Industrial Engineering, University of Salerno, 84084 Fisciano, Italy (e-mail: andevita@unisa.it; alerusso@unisa.it; ldibenedetto@unisa.it; arubino@unisa.it; gdlicciardo@unisa.it).

Danilo Pau is with STMicroelectronics, 20864 Agrate Brianza, Italy (e-mail: danilo.pau@st.com).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSI.2020.3011984

on a new partially binarized Hybrid Neural Network (HNN), which has been specifically developed [28]. It uses binarized weights to limit the number of physical resources needed for its HW implementation, in conjunction with non-binarized activations. With this combination, the system achieves an accuracy of 97.5% in classifying 5 human activities when data come from only one tri-axial accelerometer [28]. The high accuracy has also been favored by the introduction of a pre-processing stage (between the sensor and the HNN) to reduce the disturbance of the gravity acceleration, and make the acquisitions independent from the spatial orientation of the sensor. This stage increases the overall accuracy by about 4 percentage points, although it requires 16% only of the total physical resources thanks to a simplified algorithm and a careful HW design.

Measurements on the HAR system, implemented on a Xilinx Artix 7 FPGA [29], return a total power dissipation of 74 mW (almost all quiescent power) and the utilization of 6788 LUTs. With a maximum operating frequency of 41 MHz, it supports in real-time accelerometer with an Output-Data-Rate (ODR) up to 3.2 kHz. Synthesis with TSMC LP-HVT CMOS 65 nm technology returns a dynamic and static power dissipation of 2.6 μW/MHz and 5.3 mW (almost all for leakages), respectively, an area occupation 0.2 mm^2, and a maximum operating frequency of 105 MHz, which support in real-time ODR up to 8.7 kHz. All the above results overcome the state-of-the-art for this kind of systems.

The remaining of the article is organized as follows: the most relevant related works on the topic are reported in section II; section III describes the proposed models; design choice and architecture of the HW accelerator are discussed in sections IV; implementation results and comparisons with the state-of-the-art are discussed in sections V; section VI concludes the paper.

II. Related Works

The proposed HAR system, for the first time in the literature, aspires to such level of compactness to be embeddable in the sensor auxiliary circuitry. Other published solutions are far from the above target, and very few of them present the necessary trade-off between high accuracy level and low area and power, required to be effectively employed in handheld/wearable devices. In [30] an FPGA-based wearable System on Chip (SoC) for HAR is proposed. A pre-processing phase is implemented in SW and run on an embedded soft-processor, while the classification is performed using an ANN model implemented with FPGA. Input data come from 3 IMUs and 1 Heart Rate (HR) sensor, and 7 activities are classified from the PAMAP2 dataset [31]. The system achieves an accuracy of 89.7% with a power consumption of 224 mW. In [32], SensorNet Convolutional Neural Network (CNN) is used to classify multimodal time series signals. When used for HAR purposes, input data come from 3 IMUs and 1 HR sensor, and an accuracy of 98.0% is achieved when 12 activities from PAMAP2 dataset are classified. A custom HW architecture has been designed and implemented with both FPGA and CMOS 65 nm standard cells, obtaining a power consumption of 116 mW and 18.5 mW, respectively.

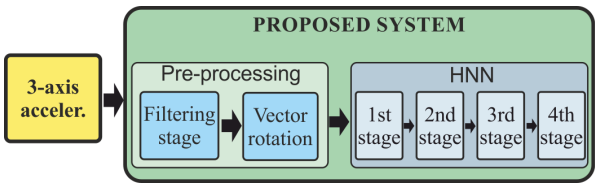

Fig. 1. Block diagram of the HAR system. The pre-processing module is important for the overall accuracy. Blue boxes have been implemented in HW.

The above values show the impact of the sensors on the total energy budget and justify our design constrains of using only one inertial MEMS [33]. A single accelerometer is used in [34], where a dedicated HW-based HAR system for smart military wearables is proposed. A Multilayer Perceptron (MLP) algorithm achieves an accuracy of 94.6% in classifying 5 military activities, and it is implemented with FPGA, where it performs the classification in 270 ns with a power consumption of 241 mW. Based on our knowledge, the only HAR system employing BNN is the Binarized Long Short-Term Memory Recurrent Neural Network (B-LSTM-RNN) in [35], which achieves a recognition accuracy of 90%, however details about its implementations are completely absent. Although in very different contexts from HAR, recent literature confirms that binarization is a proven and widely used approach, even in high performance image processing applications. The CNN in [36] is an inference accelerator for NN implementing binary weights and fixed-point activations. It exploits voltage scaling and latch-based standard-cell memories to meet a superior energy efficiency at the cost of area. Both approaches badly fit in the low frequency HAR context. Indeed, in this case static power due to leakages, and hence to area, is the main limiting factor. Thus, high threshold voltage devices are desirable to reduce the leakages, but they limit the advantages of voltage scaling. The XNOR network accelerator embedded in a microcontroller unit in [37] achieves state-of-the-art results in terms of area and energy budget, at the cost of a fully binarization of weights and activations. As will be shown in the following, this approach does not provide a suitable level of accuracy for HAR, which is the main reason why we propose a partial binarization and we introduce the pre-processing stage. The fully binarization and the binarization of the inputs from an image sensor, used in the visual system in [38], have been shown unfeasible for HAR, because of the significant accuracy loss. In particular, as will be shown ahead, the pre-processing module requires operands that are coded with longer codewords than the 16 bits used in the input layer of the HNN. For a wider and more complete overview of the literature related to binarized NN, readers could refer to review in [23].

III. Underlying Models

According to the scheme in Fig. 1, the proposed HAR system processes data from a single tri-axial accelerometer and classifies them in five classes (stationary, walking, running, biking, driving) through the pre-processing and the HNN stages [39], [27]. In the following sub-sections, the methods underlying the two stages are described.

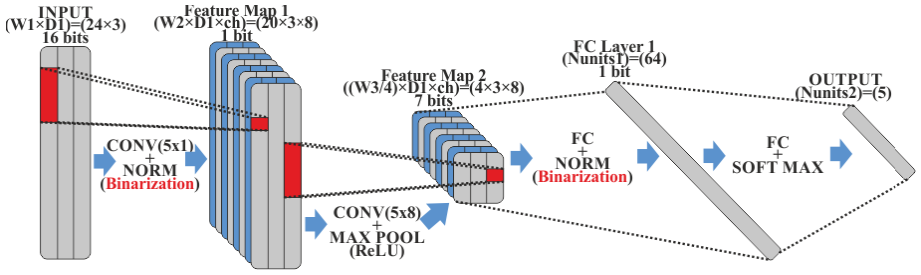

Fig. 2. Architecture of the Hybrid Neural Network. The binarization of the output activations is marked by the word "Binarization" in red. A 16-bits fixed point format is used as input. The minimum word-length required to properly represent the data is reported for each stage, along with the dimension of each feature map.

A. Hybrid Neural Network

The classification task is performed by means of the HNN schematized in Fig. 2. It is a CNN composed of seven layers with two convolutional (CONV) layers and two Fully Connected (FC) layers. The HNN structure has been started by taking inspiration from [24], and defined by trial and error procedures in order to find the topology and the set of hyperparameters that allowed to maximize the accuracy. Two CONV layers have been introduced, which operate as hierarchical extractors. In particular, the first one processes the input data, coded with high precision, by a Haar filter to extract invariant multi-spectral features. The NORM layer has been introduced to shrink the dynamic range of the activations, in order to prevent the overfitting. The Max-Pool layer operates as a conventional non-linear decimation filter. The FC layers select features by aggregator filters. Two FC layers are used, one before and one other after the binarization. All the weights have been reduced to $+1$ or -1, in order to reduce the memory requirement for coefficients, as well as the complexity of the datapath circuitry, which employs ADD/SUB operators in place of fused Multiply-Accumulate (MAC) thanks to the absence of complete convolutions. The output activations have been binarized only in those layers having the greatest demand of arithmetic operators or the highest memory requirement, as detailed in the following. The sign function is used as activation in binarized layers:

$$y = \mathrm{sgn}(x) = \begin{cases} -1 & if\ x < 0 \\ +1 & if\ x \geq 0 \end{cases} \tag{1}$$

For non-binarized layers the Rectified Linear Unit (ReLU) has been employed:

$$y = \mathrm{ReLU}(x) = \begin{cases} 0 & if\ x < 0 \\ x & if\ x \geq 0 \end{cases} \tag{2}$$

According to the scheme of Fig. 2, the HNN is composed of seven layers, which, from a functional point of view, can be grouped in four sequential stages. The computational complexity of each layer is reported in Table I in terms of required math operations and memory requirements. The inputs of the HNN are the components of the acceleration vector, pre-processed as shown in the following paragraph.

- The first stage is made up of a CONV layer and a normalization layer. The input window is made up of W1 $= 24$ samples for each one of the 3 axes, namely its dimensions are W1$\times$D1 $= 24 \times 3$ resulting in 72 samples. Inputs to the first layer are not binarized, and 16 bits are used for its representation, compatible with the maximum value provided by the sensor. The CONV layer applies a set of ch $= 8$ filters (each one represents a channel) with length 5 on the inputs, thus producing 8 different outputs per axis. The number of outputs per axis and per channel is equal to W2 $= 20$, due to the absence of zero-padding. Thus, the dimensions of the output activations of the first CONV layer are W2$\times$D1$\times$ch $= 20 \times 3 \times 8$ (480 samples). In the normalization layer each sample is scaled by a factor p, and a mean value m is subtracted. Values for p and m are learned during the training phase. In the first stage the output activations are binarized, thus each activation can be represented by 1 bit.
- The second stage is made up by a CONV layer and a Max-Pool layer. In this case, the input activations are composed by 8 channels that have been binarized considering that most of the operations are performed in this stage (Table I). As for the first stage, each axis is processed separately. The CONV layer applies a set of 8 filters of size 8×5, thus the output activations for this layer have dimensions W3$\times$D1$\times$ch $= 16 \times 3 \times 8$ (384 samples). The Max-Pooling has size 4×1, namely its output activations have dimensions W3/4$\times$D1$\times$ch $= 4 \times 3 \times 8$ (96 samples). Successively, the ReLU activation function is applied.
- The structure of the third stage is similar to the first one, but in place of the CONV layer there is a FC layer made up of Nunits1 $= 64$ neurons. Each neuron performs the dot product between the input activations vector and a weight vector. Values for weights are learned during the

TABLE I
HYBRID NEURAL NETWORK COMPLEXITY

Layer	Parameters [bytes]	Op. per window
Conv1	5	2400 ADDs
Norm1	16	480 SUBs
Conv2	40	15360 ADDs
Max Pool	0	576 SUBs + COMP
FC1	768	6144 ADDs
Norm2	128	64 SUBs
FC2	40	320 ADDs

TABLE II
CONFUSION MATRIX OF THE HNN

Actual Activity	Predicted Activity				
	Stationary	*Walking*	*Running*	*Biking*	*Driving*
Stationary	**98.383**	0.000	0.000	0.013	1.617
Walking	0.000	**99.280**	0.411	0.309	0.000
Running	0.000	2.531	**97.175**	0.220	0.000
Biking	2.538	0.887	0.000	**94.924**	1.651
Driving	0.000	0.000	0.000	2.199	**97.801**

training phase for each neuron. Even though weights are binarized, the parameters needed for this stage require most of the memory size (Table I). Considering that the input activations of the second stage are binarized, 7 bits are required to cover all the possible values for the output activations of the second and third stages.

- The last stage of the HNN is made up of a FC layer and a SoftMax classifier. The output of the latter stage represents the probability of belonging to each class, therefore the number of neurons corresponds to the number of the considered classes, Nunits2 $= 5$.

The HNN model has been built and trained using Lasagne [40]. The method in [24] has been used to train the network. For this purpose, a custom dataset of tri-axial accelerations has been created, which is composed of 1,443,958 samples for training and 922,287 for testing. By using a single precision floating point coding for the activations, the HNN achieves an average accuracy of 97.51% and the best validation error of 5.98%, whereas the accuracy measured using Courbariaux models [20] on the same dataset ranges between 54.83% to 76.27%, and the validation error rate does not fall below 16%. The confusion matrix of the proposed HNN is reported in Table II. In order to validate the obtained results and create a comparable set of results, training and testing have been repeated with the PAMAP2 [31] dataset. To preserve the number of output classes, we choose 5 activities among the 12 possible standard activities (standing, walking, running, cycling, rope jumping). In this dataset, data from tri-axial accelerometers in 3 different body positions are available, chest, hand, and ankle. For each of them, accelerometers with 2 different ranges are available (6g and 16g). A window of 5120 samples has been considered (about 50 s of data). Data are pre-processed according to the proposed model. Results are summarized in Table III. The overall mean accuracy is 93.67%, however best result is obtained by the hand accelerometer with an average accuracy of 99.968%. The higher accuracy in this case is due to the pre-processing stage which mitigates gravity acceleration that largely affect the complex hand movements. The large difference in accuracy reported in Table III, especially for ankle and chest position with sensor range equal to 6g, is a known issue of the PAMAP2 dataset, i.e. the saturation of the measured acceleration when the sensor range is 6g.

In order to show the advantage in term of energy-accuracy ratio, the proposed HNN has been compared with a fully binarized version (BNN) and an 8-bit CNN quantized version. The BNN has been built by binarizing all the weights and activations in Fig. 2, while weights and output activations have been quantized to 8-bit for the CNN. These quantized models have been trained by using the TensorFlow Lite libraries [41], without data pre-processing. Results in terms of number of MAC operations, accuracy and memory requirements are shown in Fig 3. MACs have been divided in three categories having a very different complexity and corresponding HW implementation: *full precision*, which requires 8 bits multiplications and additions; *fully binarized*, which does not require multipliers (MACs can be implemented by XNOR-popcount operations); *weight binarized*, with binarized weights but full precision input activations, for which multipliers are not required, but XNOR-popcount cannot be used. Fig. 3 shows that the accuracy of HNN, 93.11%, is significantly higher than 84.25% of BNN, and it reaches 99.45% at 8-bits. On the contrary, the memory amount required to store all the parameters of the 8-bit CNN model is 6.73 KB, namely 6.9× higher than the BNN and HNN counterparts, both requiring 0.97 KB. Moreover, the 8-bit CNN requires that all the MAC operations be full precision, while for HNN about 65% of MAC are fully binarized and the remaining is weight binarized. The number and the category of MACs are good indicators of the HW resources required for the NN implementation, and this is strictly related to the required energy budget as will be shown ahead. Therefore, results show that the HNN represents a good energy-accuracy trade-off, which is very suitable for HW implementation. The above results also justify the introduction of the pre-processing stage to compensate for the accuracy loss with respect to a non-binarized NN. As it will be shown in the following, pre-processing requires low additional HW resources compared to those of the overall system.

TABLE III
ACCURACY FOR DIFFERENT TYPES OF ACCELEROMETERS

Sensor Position	Sensor Range	Accuracy
Ankle	±6g	80.436%
Ankle	±16g	97.396%
Hand	±6g	99.968%
Hand	±16g	99.921%
Chest	±6g	89.174%
Chest	±16g	95.101%

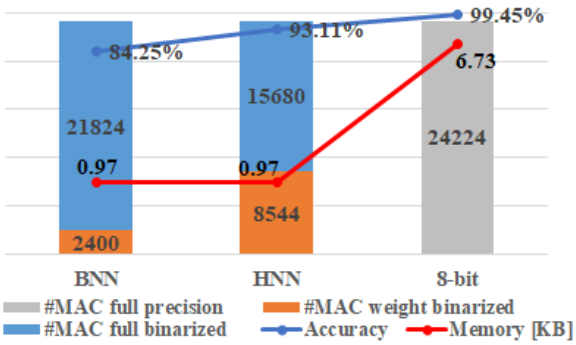

Fig. 3. Comparisons between the proposed HNN, a fully binarized NN (BNN) and a 8-bit non-binarized NN, in terms of MAC operations, memory requirements and accuracy. MAC operations have been divided in three categories to account for the very different complexity of a real MAC operator, a MAC without complete multipliers and that implementable with a simple XNOR-popcount.

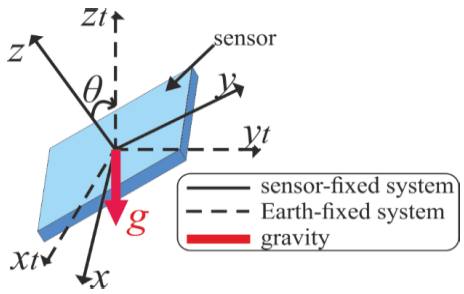

Fig. 4. Representation of the different reference frames involved in the pre-processing operations. The gravity vector is shown in red and is used to define the Earth-fixed reference frame $\{x_t, y_t, z_t\}$, to which the sensor-fixed reference frame {x, y, z} is rotated.

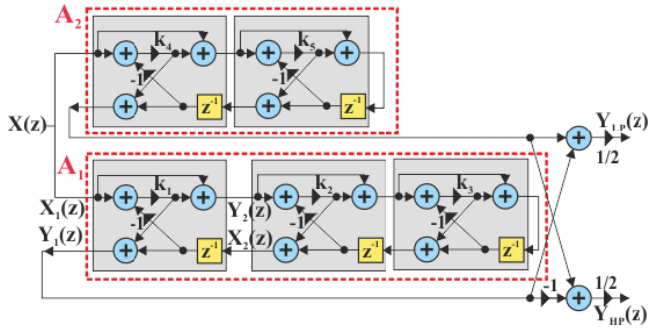

Fig. 5. Calculation scheme of the IIR Butterworth filter using a Coupled All-Pass structure. Y_{LP} and Y_{HP} are the LP and HP components, respectively. They are obtained thanks to the combination of the two all-pass filters A_1 and A_2.

B. Pre-Processing Operations

As schematized in Fig. 4, a tri-axial accelerometer provides the components of the measured acceleration along the three axes (x, y, z) composing the *frame fixed* to the sensor. This could be a serious problem when the sensor is embedded in float devices since consecutive acquisitions could not be directly compared, as they could belong to different frames. As a direct consequence, the contribution of the gravity acceleration, responsible of significant inaccuracies, cannot be easily removed. Conventional methods, indeed, are based on the trivial cancellation of the mean value calculated over an observation window of continuous samples, with the assumption that gravity is responsible for a constant acceleration component [42]. Such methods poorly adapt here, because gravity must be mandatorily removed in real-time from each sample to guarantee correct operations of the HNN in several applications. Therefore, by exploiting the method presented by the authors in [43], [44], the proposed HAR system introduces a pre-processing stage composed, in turn, by the *vector rotation* and *filtering modules*. The first one rotates each acquired vector to a unique Earth-fixed reference frame (x_t, y_t, z_t), determined by the gravity acceleration vector. The latter is extracted from each acquisition by the *filtering module*. Both filtering and vector rotation modules have been specifically designed to reduce the computational and implementation complexity of the pre-processing stage.

1) Filtering: Gravity acceleration is estimated from the low-frequency component of the acquired accelerations [45]. Considering that human activity has operation frequencies in the order of some hertz, a sharp filter is needed. Therefore, an IIR Butterworth filter with order 5 and cutoff frequency of 0.4 Hz has been specifically designed. Since we need of both the low-pass and the high-pass components of the inputs, the filter has been implemented with the Coupled All-Pass (CA) structure [46] depicted in Fig. 5. In a CA filter, the low-pass (LP) transfer function (TF) of an odd order Butterworth, Chebyshev or elliptic digital filter can be expressed as the sum of two all-pass TFs, $A_1(z)$ and $A_2(z)$:

$$Y_{LP}(z) = \frac{A_1(z) + A_2(z)}{2} \tag{3}$$

Moreover, if we consider a new TF, $Y_{HP}(z)$ defined as:

$$Y_{HP}(z) = \frac{A_1(z) - A_2(z)}{2} \tag{4}$$

the following relationship holds:

$$\left|Y_{LP}\left(e^{j\omega}\right)\right|^2 + \left|Y_{HP}\left(e^{j\omega}\right)\right|^2 = 1 \tag{5}$$

Equation (5) reveals that the frequency responses of $Y_{LP}(z)$ and $Y_{HP}(z)$ are complementary [46]. Therefore, the bandwidth of $Y_{LP}(z)$ coincides with the stopband of $Y_{HP}(z)$ and vice-versa and, hence, $Y_{HP}(z)$ is a high-pass TF. The Coupled All-Pass (CA) structure schematized in Fig. 5 [46] has been preferred for its regular structure, which enables an iterative design around a simple fundamental cell containing 3 additions and 1 multiplication, evidenced by the grey boxes in the figure. In fact, considering a generic m^{th} order all-pass TF, $A_m(z)$:

$$A_m(z) = \frac{d_m + d_{m-1}z^{-1} + \ldots + d_1 z^{-(m-1)} + z^{-m}}{1 + d_1 z^{-1} + \ldots + d_{m-1}z^{-(m-1)} + d_m z^{-m}} \tag{6}$$

we can express it in terms of a $(m-1)^{th}$ order all-pass TF, $A_{m-1}(z)$:

$$A_m(z) = \frac{k_m + z^{-1}A_{m-1}(z)}{1 + k_m z^{-1} A_{m-1}(z)} \tag{7}$$

where $k_m = A_m(\infty) = d_m$. By repeating this argument recursively, $A_{m-1}(z)$ can be realized by "extracting" a two-pair constrained by a TF $A_{m-2}(z)$ and so on. This approach leads to the realization of $A_m(z)$ as a cascaded connection of m equal two-pairs. With reference to Fig. 5, the equations for each elementary cell are the following:

$$\begin{cases} Y_1 = V_1 + z^{-1}X_2 \\ Y_2 = V_1 + X_1 \\ V_1 = k_m\left(X_1 - z^{-1}X_2\right) \\ X_2 = A_{m-1}(z)\, Y_2 \end{cases} \tag{8}$$

The advantages of the proposed filter with respect to other known solutions in terms of the physical resources needed for their implementation is shown in Table IV.

2) Reference Frame Rotation: To represent the human motion acceleration in the Earth-fixed reference frame (Fig. 4), a 3D rotation must be performed. Many methods exist in the literature to perform this operation, such as the Euler angles [47], the Rodrigues' rotation formula [48], and quaternions [49]. However, all these methods require the computation of many complex arithmetic functions, such

TABLE IV
HW RESOURCES FOR DIFFERENT FILTER STRUCTURES OF ORDER M

Filter Structure	#Multipliers	#Registers	#Adders
II Direct Form Transposed	2m+1	2m+1	2m
Cascaded Form	2m+1	2m+1	2m
Coupled All-Pass	**m**	**m**	**3m+2**

TABLE V
NUMBER OF OPERATIONS FOR DIFFERENT REFERENCE FRAME TRANSFORMATIONS ALGORITHMS

Methods	#Add	#Mult	#Func	Type of functions
Euler Angles	12	27	13	arctg, div, sqrt, sin, cos
Rodrigues'	18	27	7	sqrt, div, arcsin, sin, cos
Quaternions	34	54	9	sqrt, arcsin, sin, cos, div
Proposed	**13**	**25**	**4**	**sqrt, div**

as trigonometric functions, square roots, and normalization. Their implementation generates cumbersome HW. Thus, the HW-friendly algorithm explained in [44] has been used in the proposed system. It avoids the computation of trigonometric functions and requires a lower number of operations compared to other methods, as shown in Table V. The principal steps of the implemented method are summarized below. Inputs from the filter are the gravity vector $\vec{g} = (g_x, g_y, g_z)$ and the human motion acceleration $\vec{v} = (v_x, v_y, v_z)$. They are represented in the sensor fixed reference frame defined by the unit vectors $\hat{x} = (1, 0, 0)$, $\hat{y} = (0, 1, 0)$ and $\hat{z} = (0, 0, 1)$. The resulting human motion acceleration, rotated in the Earth-fixed reference frame, (v_{xr}, v_{yr}, v_{zr}), is calculated as:

$$v_{xr} = \vec{v} \cdot \hat{x}_t, \quad v_{yr} = \vec{v} \cdot \hat{y}_t, \quad v_{zr} = \vec{v} \cdot \hat{z}_t \tag{9}$$

with:

$$\begin{cases} \hat{x}_t = \vec{x}_u + \vec{x}_n = \left(-\frac{|g_x| g_x g_z}{\|\vec{g}\| g_{xy}^2} + \frac{g_y^2}{g_{xy}^2}, -\frac{|g_x| g_y g_z}{\|\vec{g}\| g_{xy}^2} - \frac{g_x g_y}{g_{xy}^2}, \frac{|g_x|}{\|\vec{g}\|} \right) \\ \hat{y}_t = \hat{z}_t \times \hat{x}_t = \left(\frac{a_y g_z - a_z g_y}{\|\vec{g}\|}, \frac{a_z g_x - a_x g_z}{\|\vec{g}\|}, \frac{a_x g_y - a_y g_x}{\|\vec{g}\|} \right) \\ \hat{z}_t = -\frac{\vec{g}}{\|\vec{g}\|} = \left(-\frac{g_x}{\|\vec{g}\|}, -\frac{g_y}{\|\vec{g}\|}, -\frac{g_z}{\|\vec{g}\|} \right) \end{cases}$$

where:

$$\begin{cases} \hat{x}_t = (a_x, a_y, a_z) \\ g_{xy} = \sqrt{g_x^2 + g_y^2}, \quad \|\vec{g}\| = \sqrt{g_{xy}^2 + g_z^2} \\ \vec{x}_u = (\hat{x} \cdot \hat{u})\hat{u}, \quad \vec{x}_n = \left\| \hat{x} - \vec{x}_u \right\| (\hat{z}_t \times \hat{u}) \\ \hat{u} = \frac{\vec{g} \times \hat{z}}{\left\| \vec{g} \times \hat{z} \right\|} = \left(\frac{g_y}{g_{xy}}, -\frac{g_x}{g_{xy}}, 0 \right) \end{cases}$$

The derived calculation scheme, shown in Fig. 6, is implemented by a basic building block, composed of 3 multipliers and 2 adders, operating in an iterative fashion [40]. All the divisions have been conveniently fused in 3 operators and moved to the end of the scheme.

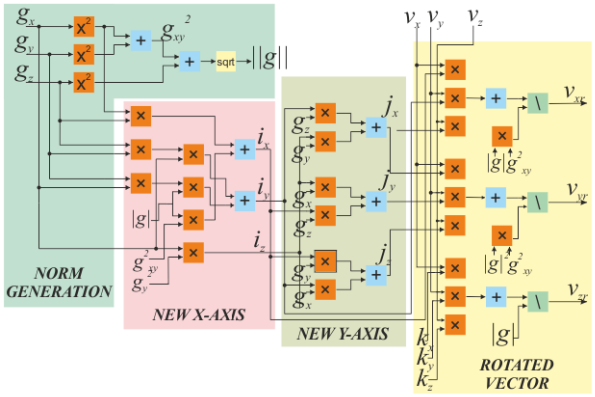

Fig. 6. Calculation scheme of the modified reference frame rotation algorithm. All the divisions are shifted at the end of the algorithm.

C. Accuracy of the Pre-Processing

Although the entire model has been developed by using single precision floating point (FP) arithmetic, for non-binarized values, a fixed point (FI) coding has been preferred for both the filter and the rotation module, to highly reduce the number of physical resources required for their circuital implementation. Tests on the modules have been conducted to select the appropriate codelengths and to find the most effective trade-off between the minimum word-length and the resulting accuracy. With reference to the filter, the FI coding introduces an error in the representation of the coefficients, and hence poles and zeros experience variations with respect to their theoretical values. In the worst-case scenario, the filter could become instable. In Fig. 7a,b, the comparison between the ideal frequency responses of the CA filter and the quantized ones is shown for three codelengths: 20 (8.12), 24 (8.16) and 28 (8.20) bits. The frequency response obtained with a FP 64-bits coding has been considered for reference. Although the filter is stable for all the considered codelengths, the deviations of the cut-off frequency are 0.005%, 0.90% and 13.75% ranging from 20 to 28 bits, respectively. Therefore, 24 bits have been chosen to limit the error under 1%.

With reference to the frame rotation, the largest part of the error is concentrated in the square-root calculation of the scheme in Fig. 6. To keep the error below the precision of the adopted FI coding, the square root function has been implemented by using the third-order Taylor series in (10), and it has been expanded around 11 different points. In Fig. 8, it is shown that the approximation error is always below the precision σ, which is equal to 2^{-16} for the chosen wordlength.

$$\begin{aligned} \sqrt{r} = \sqrt{1+x} = \sqrt{1+x_0} + \frac{1}{2\sqrt{1+x_0}}(x - x_0) + \\ - \frac{1}{8\sqrt{(1+x_0)^3}}(x - x_0)^2 \\ + \frac{1}{16\sqrt{(1+x_0)^5}}(x - x_0)^3 + \ldots \end{aligned} \tag{10}$$

IV. ARCHITECTURE DESIGN AND IMPLEMENTATION CHOICES

The implementation scheme in Fig. 1 is composed of the pre-processing module and the HNN accelerator. By exploiting

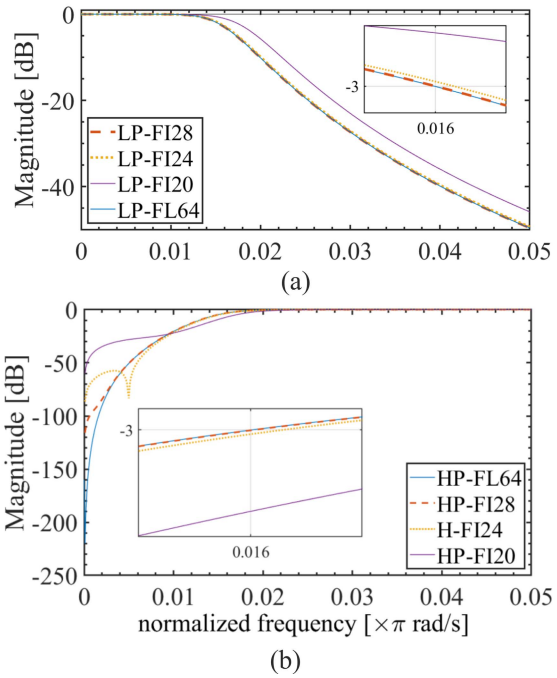

Fig. 7. (a) Comparison between ideal and FI LP frequency responses. (b) Comparison between ideal and FI HP frequency responses. In both cases, the inset shows the error in the −3dB cutoff frequency.

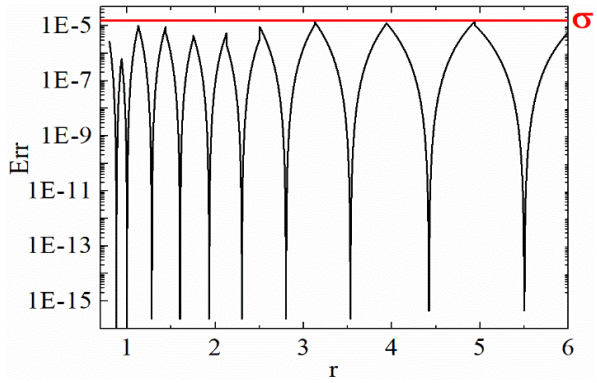

Fig. 8. Approximation error in square root function computation using a third-order Taylor series expansion over the range [0.8, 6]. The function has been expanded around 10 points: {−0.12, 0, 0.28, 0.60, 0.93, 1.30, 1.80, 2.53, 3.42, 4.50}.

the low frequencies typical of the human activities, the implementation strategy for both modules followed an extensive use of resource sharing and iterative processing schemes. This allowed to reduce the total amount of HW resources, the related area, and hence to bring down the leakage/static power dissipation, which, as will be shown, is the dominant component in the proposed HAR systems.

A. Pre-Processing Module

The filter and the module for the reference frame rotation share the circuitry schematized in Fig. 9. According to the scheme in Fig. 6, a repetitive pattern made up of 3 multiplications and 2 additions has been detected for the reference frame rotation, while 3 additions and 1 multiplication are needed for the fundamental cell of the filter. In turn, for each

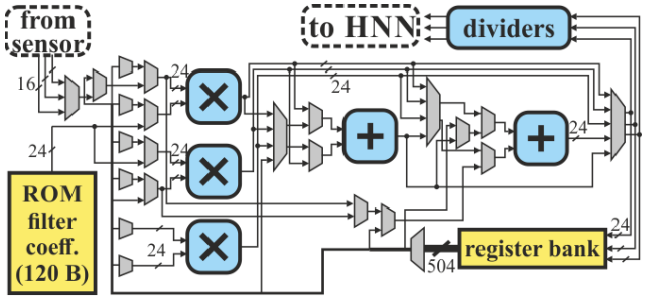

Fig. 9. Block diagram of the circuit which implements the entire pre-processing module.

operator, small and iterative topologies have been selected. Based on that, the arithmetic core of the pre-processing module is composed of 3 Booth multipliers operating in parallel, 2 cascaded carry-ripple adders, and multiplexers to properly set the configuration of the module at each cycle. A 500 bytes register bank stores the intermediate results at each cycle. To further reduce the mapped resources, each multiplier is iteratively made up of a single Booth cell; therefore 12 cycles are required to calculate a product. A dedicated clock has been used for the timing of the multiplier. This faster *clk_mult* has been generated from the external oscillator (12 MHz those provided by the FPGA dev-board), while the 12× slower global clock (*clk*) has been obtained by dividing the frequency of *clk_mult*. Finally, the dividers at the end of the scheme have been implemented by a conventional iterative restoring algorithm [50]. In total, 52 *clk* cycles, equivalent to 624 *clk_mult* cycles, are required to pre-process an input sample.

B. Hybrid Neural Network Accelerator

Two aspects are critical in the HW implementation of ANNs: the large number of arithmetic operators and the allocation of a large amount of memory for storing weights and partial results, as well as the power dissipation related to the numerous memory accesses [19]. In our implementation, weight binarization has reduced the MAC operations to simple ADD/SUB operations, namely each CONV layer calculates the following quantities:

$$\sum_{i=1}^{N} w_i x_i + b = \pm x_1 \pm x_2 \pm \ldots \pm x_N + b \qquad (11)$$

where w_i and x_i are the weights and the inputs to a certain neuron, respectively, and b is the bias [51], [52]. Since the energy cost of data read/write operations from off-chip memories can be up to 200× higher than on chip data transfer [19], an effort has been done to use only on-chip memories. Two designs of the HNN accelerator are proposed: a FIFO-based design, where memories have been implemented by using distributed FIFOs and RAM has been completely avoided. A second version uses RAM to store weights and biases, while FIFOs continue to be used to store the output activations. The choice to present both solutions derive from the need to find different optimal area/power trade-off in different utilization scenario. Auxiliary circuitry of sensors, indeed, are equipped with very limited amount of RAM memory, which could be insufficient for the HAR operations and compel to the use

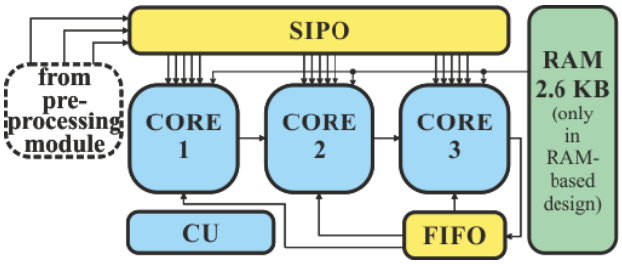

Fig. 10. Block diagram of the proposed HNN accelerator. The RAM module is present in the RAM-based design only. The structure of the cores is different for the two versions.

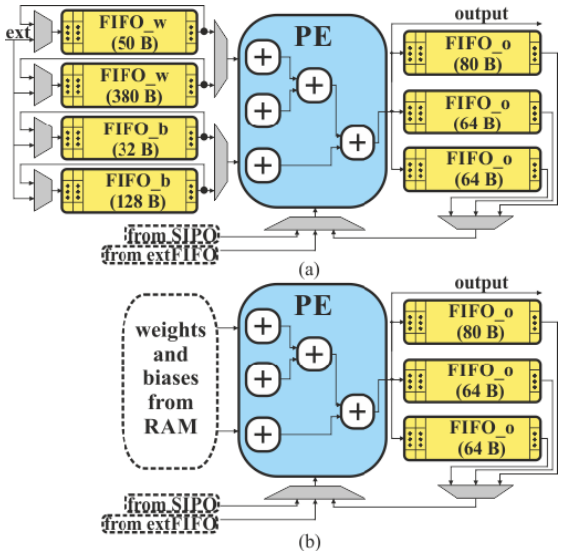

Fig. 11. (a) Block diagram of a core in the FIFO-based design. In this case weights and biases are stored in FIFO memories locally. (b) Block diagram of a core in the RAM-based design. In this case weights and biases are stored in a RAM module which is shared among the cores.

of FIFOs. In turn, FIFOs permits higher operation frequencies with respect to RAM, and the associated dynamic power scale-up with a lower slope than RAM. This make FIFOs convenient for higher frequencies applications. On the contrary, RAM could be a convenient choice for target platform such as FPGA, which could advantage of distributed memories and the lower power dissipation for data transfers, due to the locality of data. The block diagram of both HNN designs is shown in Fig. 10. The architecture exploits 3 cores, since this is the minimum number of cores which can process in parallel the 3 components of the pre-processed acceleration. In Fig. 11a and Fig. 11b, the architectures of the cores of the FIFO-based and the RAM-based HNN accelerator are detailed. Thanks to weight binarization, the processing element (PE) is a 3-levels adder tree which uses 16-bits FI arithmetic in both cases. The first level of the adder tree is made up of 3 adders, so that a dot product between vectors of length 5 can be performed in one cycle, and a bias or a result from the previous cycle can be summed up as well.

1) FIFO-Based HNN Accelerator: In the FIFO-based HNN accelerator, each core embeds 800 bytes of FIFO memories in which weights, biases and partial results are stored. Each core locally stores all the parameters needed to run the model. Also, output activations from CONV layers are locally reused in each core. Thus, the design takes advantage of a "flattened" memory hierarchy, where there is no need to execute high cost access operations to higher levels in a memory hierarchy. To make this possible, FIFOs must be initialized during the system start-up by an external data stream. Successively, FIFOs work as a circular buffer, carefully managed by a Control Unit (CU). In particular, in the design of Fig. 11a, the FIFO_w structures store the weights of the model, whereas the FIFO_b structures store the biases. Two different "FIFO_w" and "FIFO_b" modules were needed in each core and used when the circuit implements a CONV layer or a FC layer, respectively. Indeed, CONV layers must be processed 16 times to get a new input for the FC layers. Thus, considering that in FIFO structures we cannot have random accesses to the memory locations, we should have had to swipe all the weights of the CONV and the FC layers even when the latter would have not been useful. This would have been a drawback for the design, because the number of weights of the FC layers requires the 77% of the total memory required to store the network parameters, as reported in Table I. In particular, considering that a weight in the HNN is represented by a single bit, and that each CONV layer has a filter with dimension proportional to 5 (5 or 5×8), FIFOs for CONV layers have dimensions 80×5 bits, while those for FC layers are 608×5 bits. The same applies for "FIFO_b" but, considering that biases are coded with FI 16-bits, FIFOs requires 16×16 bits and 64×16 bits, respectively. "FIFO_o" stores the output activations of each layer. Each one of the above output FIFO is divided in up to 5 blocks, in order to provide up to 5 different output activations in parallel to the PE, designed to perform a dot product between vectors of length 5 in one cycle. In Fig. 11a, the 3 "FIFO_o" memories store the output activations of the first stage, the second CONV layer and the Max-Pool layer, respectively. Each axis is processed separately in CONV layers, thus the memory for the output activations is locally associated to each core. As shown in Fig. 10, a unique external FIFO memory is also used to store the output activations of the first FC layer, since in this case, all the input activations from the previous layer cannot be separated. Considering that the scheme in Fig. 9 iteratively implements all the HNN layers, the complex signal routing is managed by the devoted CU in Fig. 10, in turn implemented with a Finite State Machine (FSM) having a state for each layer.

2) RAM-Based HNN Accelerator: The RAM-based design instantiates a RAM to store weights and biases in place of FIFOs (Fig. 10). This choice is advantageous in terms of power dissipation since it avoids the data shifts that FIFOs do at each read operation, although the highest advantage is obtained with the availability of on-chip distributed RAM typical of FPGAs. The cores of the RAM-based HNN accelerator in Fig. 11b have a similar structure to the FIFO-based ones, but a RAM module of 31×696 bits reduces the FIFOs dimensions to 200 bytes. The most significant 15 bits of each word of the RAM are used to store weights, therefore again each core receives 5 binarized weights at each cycle. The remaining 16 bits are used for the biases. The proposed architecture has been prototyped with a Xilinx Artix-7 FPGA and, for ease of comparison also with

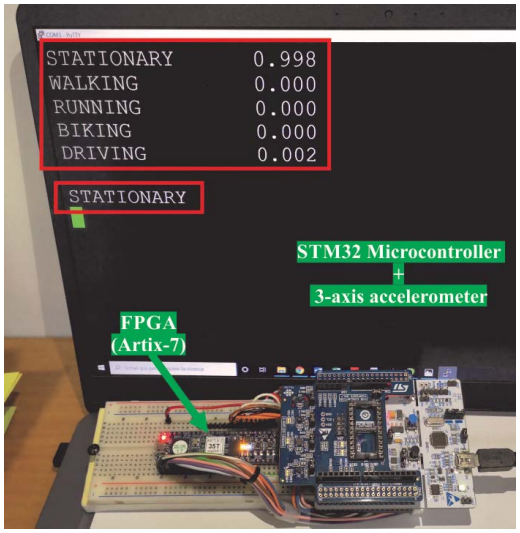

Fig. 12. FPGA-based demo board. The scores for each one of the 5 classes and the consequent classification are printed to video in real-time.

standard cells (std_cells) implementation. In the latter case, a larger SRAM of 32×704 bits has been instantiated due to the limitations of the memory compiler.

V. Synthesis and Implementation Results

The proposed design has been implemented on the Xilinx Artix-7 (xc7a35tfgg484-1) FPGA by using the Xilinx Vivado environment and synthesized with TSMC CMOS std_cells by using the Cadence toolchain.

A. FPGA Results

The HAR system prototype is shown in Fig. 12. A small Digilent CMOD A7-35T [29] has been used to implement the entire circuitry, while the X-NUCLEO-IKS01A1 [28] equipped with the LSM6DSO IMU is used as tri-axial accelerometer. The STM32F411RE [53] microcontroller is used to manage the data transfer between IMU and FPGA and to display the processed results. In Table VI the results of the FPGA implementation are summarized for both the FIFO-based and the RAM-based designs. The FPGA takes advantage of the presence of RAM since the LUT utilization is reduced by about 3% with respect to the FIFO-based counterpart, namely from 31.7% to 28.8% of the total number of the available LUTs on the FPGA. Analogously, the FF utilization is reduced by about 2.3% since 1017 FFs are used to store weights and biases of the HNN. The maximum clock frequency for both designs is 41 MHz, corresponding to a maximum Output Data Rate (ODR) of the sensor of 3.2 kHz. However, when the ODR of the sensor is set to 25 Hz, normally used in HAR systems, considering that 12600 clock cycles are needed to process a data, the minimum clock frequency required for real-time operation is 315 kHz, indicated as the operating frequency (OpFreq) in Table VI. At this frequency, considering that 16 input samples are required to obtain a classification, the delay (Delay@OpFreq) is equal to $16 \times (1/25\ \text{Hz}) = 640$ ms. As shown in Table VI, the value is higher than in [32] and [34]. However, this is justified by the higher operating frequencies used in those works, which is not actually required in the proposed HAR system. Vivado power tool, set-up at high level of confidence by Switching Activity Interchange Format (SAIF) generated from post-implementation simulations, returns for both designs a total power consumption of 72.04 mW at the OpFreq. This is almost all composed of static power, equal to the quiescent power dissipation of the FPGA. Dynamic power is under the sensitivity of the tool which returns a generic <1 mW. Measurements on the CMOD board returns a maximum current of about 100 mA in both cases, which is obviously increased by the additional components of the board. Therefore, for the FPGA implementation, there are not significant differences between the two designs and the RAM-based design could be preferred since it takes advantage of the primitives of the FPGA, while LUTs and FFs can be saved for other purposes. In Table VI the proposed design has been compared to state-of-the art literature [32], [34], oriented to custom HW implementation. Accuracy is significantly higher than [34] and quite lower than that in [32], although hand movements with PAMAP2 returns much better results. However, 3 accelerometers and 1 heart-rate monitor are used as input sensors in [32], whereas only 1 tri-axial accelerometer is used in the proposed system. This should be considered in the economy of the whole system both in terms of power consumption and area occupation. A lower number of resources is required by our designs, although we do not instantiate DSP modules in order to provide results that are independent from the specific target platform, and for a fair comparison with std_cell implementations. The total RAM requirement for our RAM-based design is equivalent to 1 BRAM and 0 for the FIFO-based, while in [32] a significant number of BRAMs has been used. The power consumption has

TABLE VI
FPGA Implementation Results

	FIFO	***RAM***	*Jafari [32]*	*Gaikwad [34]*
Platform	Artix-7	Artix-7	Artix-7	Artix-7
Accuracy	97.5%	97.5%	98.0%	94.6%
Dyn. Power [μW/MHz]	137	134	460	-
Static Power [mW]	72	72	71	-
Total Power @ OpFreq [mW]	72.04	72.04	116	241
# of slices	2093	1856	982	-
# of LUTs	6601	5988	-	3466
# of FFs	5272	4299	-	569
# of DSPs	0	0	3	81
# of BRAMs	0	1	14	0
Max Frequency [MHz]	41	41	-	-
Max Sensor ODR [kHz]	3.2	3.2	-	-
Delay @ OpFreq [ms]	640	640	14.8	$2.7\text{x}10^{-4}$

TABLE VII
STD_CELLS SYNTHESIS RESULTS AND COMPARISONS

	FIFO-based design		*RAM-based design*	Jafari [32]
Technology	CMOS 65 nm LP HVT	CMOS 90 nm	CMOS 90 nm	CMOS 65 nm
Dyn. Power [μW/MHz]	2.6	3.1	8.8	111
Leak.Power [mW]	$5.4x10^{-3}$	2.5	1.4	7.4
Total Power @ OpFreq [mW]	$6.3\ x10^{-3}$	2.5	1.4	-
Total Power @ 67 label/s [mW]	-	2.54	1.52	18.5
Area [mm^2]	0.20	0.36	0.39	0.40
Max Frequency [MHz]	105	158	97	857
MaxThroughput [label/s]	-	784	480	574
Delay @ 67 label/s [ms]	-	14.9	14.9	14.8
Energy [μJ]	-	38	23	274
Max Sensor ODR [kHz]	8.7	12.5	7.7	-

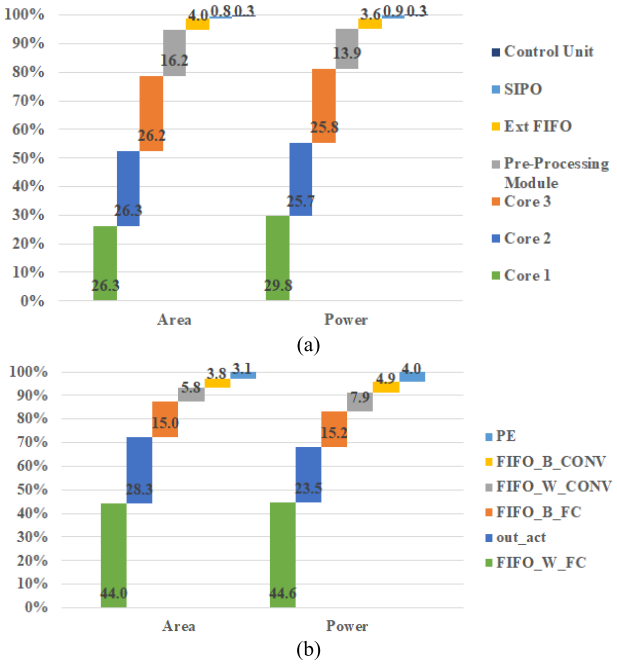

Fig. 13. Breakdown of area occupation and power consumption of: (a) various sub-modules of the FIFO-based accelerator; (b) components of one core of the FIFO-based HNN accelerator.

been compared at the OpFreq of each system. The proposed design shows a reduction of the power consumption of 37% and 70% with respect to [32] and [34] respectively. However, to make comparisons independent from the OpFreq, the normalized dynamic power consumption has been compared, where the proposed designs achieve a reduction of the 70% with respect to [34].

B. Standard Cells Results

In Table VII the results of synthesis with TSMC CMOS 90 nm std_cells are summarized for both the FIFO-based and the RAM-based designs and compared with the solution in [32], which only presents ASIC results. The power consumption has been estimated by extracting Value Change Dump (VCD) files from post-synthesis simulations using Standard Delay Format (SDF) files. The power consumption has been estimated using Cadence Joules. The dynamic power consumption of the RAM-based design is 2.8× higher with respect to the FIFO-based design, while the leakage power is 1.8× lower. However, the dynamic power consumption is negligible at the OpFreq. Therefore, despite of the shifting of the data in the FIFOs does not represent an issue for the dynamic power consumption, the best solution to reduce the power consumption at the considered frequencies is the RAM-based design. On the contrary, being the memory distributed in the FIFO-based design, the maximum frequency is 1.6× higher than the one of the RAM-based design. This could be considered for applications in which a high throughput is the first specification. Moreover, to verify the scaling capabilities and the actual impact of leakages, the proposed design has been synthesized with TSMC CMOS 65 nm low-power (LP) high-voltage-threshold (HVT). The HVT feature allows to strongly reduce the leakage power, at the cost of reduced speed. Results are reported in Table VII. Unfortunately, the lack of memory compiler for the 65 nm technology prevented the possibility to synthesize a RAM-based design with the more shrinked technology. Results show that the power consumption is only 6.3 μW at the OpFreq, that is 3 orders of magnitude lower than the above results. Despite this, the 86% of the total power is leakage power. The overall area occupation is 0.20 mm^2. A detail of the various components of the design is shown in Fig. 13a and Fig. 13b. In Fig. 13a is shown that most of the area is required for the HNN accelerator module. In particular, each core occupies the 26% of the architecture, whereas the pre-processing module occupies the 16% of the total area. The remaining area is mainly used for the external FIFO (see Fig. 10), the SIPO and the CU. Also, in Fig. 13b the breakdown of the various modules in each core is shown. The 96.9% of the area occupation in each core is due to FIFO memories, and only the remaining 3.1% is required to implement the PE. In fact, thanks to weight binarization, the PE has been implemented avoiding multipliers, whose implementation requires large and power-hungry circuits. In Fig. 13a,b also a breakdown of the power consumption has been reported. Considering that power dissipation is mostly due to leakages, power consumption breakdown exactly follows the one of the area occupation. In Table VII the proposed design has also been compared to [32]. To have a fair comparison of the power consumption, we have considered results at the same throughput. In [32] a throughput of 67 label/s is achieved at a clock frequency of 100 MHz. In the proposed solution, 202k clock cycles are required to produce a label. Thus, a lower clock frequency of (67 × 202k) Hz = 13.5 MHz is required to have a throughput of 67 label/s. At this frequency, the FIFO-based design and the RAM-based design dissipate

2.54 mW and 1.52 mW, which corresponds to a reduction of the 7.3× and 12.2× respectively with respect to [32]. The delay required to produce a label and the area occupation of the proposed designs are almost the same as in [32]. The maximum frequency in [32] is 5.4× and 8.8× higher than the FIFO-based design and the RAM-based design, respectively. Despite this, the proposed design is able to provide a 1.4× higher throughput in the case of the FIFO-based design.

VI. Conclusion

In this article, a new HNN model and a custom HW accelerator for HAR applications has been proposed. The system obtains high accuracy with low power consumption and resources compared to the state-of-the-art. Results show that the system is able to accomplish its task with a minimum power consumption of 6.3 μW and an area occupation of 0.2 mm^2, that is promising for the integration of the accelerator in the same die with the sensing element, thus realizing an AI-based edge device. Future works will be aimed to the extension of the architecture to other applications requiring high operation frequencies and the possibility to use ternary weights.

References

[1] O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 3, pp. 1192–1209, 3rd Quart., 2013.

[2] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition: A survey," *Pattern Recognit. Lett.*, vol. 119, pp. 3–11, Mar. 2019.

[3] Y. Xian, X. Rong, X. Yang, and Y. Tian, "Evaluation of low-level features for real-world surveillance event detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 3, pp. 624–634, Mar. 2017.

[4] A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," *ACM Comput. Surveys*, vol. 46, no. 3, pp. 1–33, Jan. 2014.

[5] H. Yu, S. Cang, and Y. Wang, "A review of sensor selection, sensor devices and sensor deployment for wearable sensor-based human activity recognition systems," in *Proc. 10th Int. Conf. Softw., Knowl., Inf. Manage. Appl. (SKIMA)*, Chengdu, China, 2016, pp. 250–257.

[6] B. M. Eskofier *et al.*, "Recent machine learning advancements in sensor-based mobility analysis: Deep learning for Parkinson's disease assessment," in *Proc. 38th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Orlando, FL, USA, Aug. 2016, pp. 655–658.

[7] T. R. Bennett, J. Wu, N. Kehtarnavaz, and R. Jafari, "Inertial measurement unit-based wearable computers for assisted living applications: A signal processing perspective," *IEEE Signal Process. Mag.*, vol. 33, no. 2, pp. 28–35, Mar. 2016.

[8] I. Bisio, A. Delfino, F. Lavagetto, and A. Sciarrone, "Enabling IoT for in-home rehabilitation: Accelerometer signals classification methods for activity and movement recognition," *IEEE Internet Things J.*, vol. 4, no. 1, pp. 135–146, Feb. 2017.

[9] F. Hussain, F. Hussain, M. Ehatisham-ul-Haq, and M. A. Azam, "Activity-aware fall detection and recognition based on wearable sensors," *IEEE Sensors J.*, vol. 19, no. 12, pp. 4528–4536, Jun. 2019.

[10] G. Cola, M. Avvenuti, and A. Vecchio, "Real-time identification using gait pattern analysis on a standalone wearable accelerometer," *Comput. J.*, vol. 60, no. 8, pp. 1173–1186, Jan. 2017.

[11] G. De Leonardis *et al.*, "Human activity recognition by wearable sensors : Comparison of different classifiers for real-time applications," in *Proc. IEEE Int. Symp. Med. Meas. Appl. (MeMeA)*, Rome, Italy, Jun. 2018, pp. 1–6.

[12] C. Xu, D. Chai, J. He, X. Zhang, and S. Duan, "InnoHAR: A deep neural network for complex human activity recognition," *IEEE Access*, vol. 7, pp. 9893–9902, 2019.

[13] T. Zebin, P. J. Scully, N. Peek, A. J. Casson, and K. B. Ozanyan, "Design and implementation of a convolutional neural network on an edge computing smartphone for human activity recognition," *IEEE Access*, vol. 7, pp. 133509–133520, 2019.

[14] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[15] *iNEMO Inertial Module: Always-on 3D Accelerometer and 3D Gyroscope*, STMicroelectronics, Geneva, Switzerland, Dec. 2018.

[16] V. Bianchi, M. Bassoli, G. Lombardo, P. Fornacciari, M. Mordonini, and I. De Munari, "IoT wearable sensor and deep learning: An integrated approach for personalized human activity recognition in a smart home environment," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8553–8562, Oct. 2019.

[17] J. Pagan *et al.*, "Toward ultra-low-power remote health monitoring: An optimal and adaptive compressed sensing framework for activity recognition," *IEEE Trans. Mobile Comput.*, vol. 18, no. 3, pp. 658–673, Mar. 2019.

[18] Z. Zou, Y. Jin, P. Nevalainen, Y. Huan, J. Heikkonen, and T. Westerlund, "Edge and fog computing enabled AI for IoT—An overview," in *Proc. IEEE Int. Conf. Artif. Intell. Circuits Syst. (AICAS)*, Hsinchu, Taiwan, Mar. 2019, pp. 51–56.

[19] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proc. IEEE*, vol. 105, no. 12, pp. 2295–2329, Dec. 2017.

[20] M. Courbariaux, Y. Bengio, and J. P. David, "BinaryConnect: Training deep neural networks with binary weights during propagations," in *Proc. 28th Int. Conf. Neural Inf. Process. Syst.*, Montreal, QC, Canada, vol. 2, 2015, pp. 3123–3131.

[21] H. Alemdar, V. Leroy, A. Prost-Boucle, and F. Petrot, "Ternary neural networks for resource-efficient AI applications," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Anchorage, AK, USA, May 2017, pp. 2547–2554.

[22] *IEEE Standard for Floating-Point Arithmetic*, Standard754-2008, Aug. 2008.

[23] T. Simons and D. J. Lee, "A review of binarized neural networks," in *MDPI Electron.*, vol. 8, no. 6, pp. 661–686, Jun. 2019.

[24] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks: Training deep neural networks with weights and activations constrained to $+1$ or -1," 2016, *arXiv:1602.02830*. [Online]. Available: http://arxiv.org/abs/1602.02830

[25] A. Prost-Boucle, A. Bourge, and F. Pétrot, "High-efficiency convolutional ternary neural networks with custom adder trees and weight compression," *ACM Trans. Reconfigurable Technol. Syst.*, vol. 11, no. 3, pp. 1–24, Dec. 2018.

[26] S. Yin *et al.*, "An energy-efficient reconfigurable processor for binary- and ternary-weight neural networks with flexible data bit width," *IEEE J. Solid-State Circuits*, vol. 54, no. 4, pp. 1120–1136, Apr. 2019.

[27] STMicroelectronics. (May 2015). *X-NUCLEO-IKS01A1 Motion MEMS and environmental sensor expansion board for STM32 Nucleo*. [Online]. Available: https://www.st.com/resource/en/datasheet/x-nucleo-iks01a1.pdf

[28] D. P. Pau, E. Plebani, F. G. De Ambroggi, F. Guido, and A. Bosco, "Recognition method, corresponding system and computer program product," U.S. Patent 16 189 264, Oct. 12, 2019.

[29] Xilinx. (Feb. 2018). *7 Series FPGAs Data Sheet: Overview*. [Online]. Available: https://www.xilinx.com/support/documentation/data_sheets/ds180_7Series_Overview.pdf

[30] K. Basterretxea, J. Echanobe, and I. del Campo, "A wearable human activity recognition system on a chip," in *Proc. Conf. Des. Archit. Signal Image Process.*, Oct. 2014, pp. 1–8.

[31] A. Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," in *Proc. 16th Int. Symp. Wearable Comput.*, Jun. 2012, pp. 108–109.

[32] A. Jafari, A. Ganesan, C. S. K. Thalisetty, V. Sivasubramanian, T. Oates, and T. Mohsenin, "SensorNet: A scalable and low-power deep convolutional neural network for multimodal data classification," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 66, no. 1, pp. 274–287, Jan. 2019.

[33] G. Chatterjee, L. Latorre, F. Mailly, P. Nouet, N. Hachelef, and C. Oudea, "Smart-MEMS based inertial measurement units: Gyro-free approach to improve the grade," *Microsyst. Technol.*, vol. 23, no. 9, pp. 3969–3978, Sep. 2017.

[34] N. B. Gaikwad, V. Tiwari, A. Keskar, and N. C. Shivaprakash, "Efficient FPGA implementation of multilayer perceptron for real-time human activity classification," *IEEE Access*, vol. 7, pp. 26696–26706, 2019.

[35] M. Edel and E. Koppe, "Binarized-BLSTM-RNN based human activity recognition," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Oct. 2016, pp. 1–7.

[36] R. Andri, L. Cavigelli, D. Rossi, and L. Benini, "YodaNN: An architecture for ultralow power binary-weight CNN acceleration," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 37, no. 1, pp. 48–60, Jan. 2018.

[37] F. Conti, P. D. Schiavone, and L. Benini, "XNOR neural engine: A hardware accelerator IP for 21.6-fJ/op binary neural network inference," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 37, no. 11, pp. 2940–2951, Nov. 2018.
[38] M. Rusci, D. Rossi, E. Flamand, M. Gottardi, E. Farella, and L. Benini, "Always-ON visual node with a hardware-software event-based binarized neural network inference engine," in *Proc. 15th ACM Int. Conf. Comput. Frontiers*, Ischia, Italy, May 2018, pp. 314–319.
[39] U. Maurer, A. Smailagic, D. P. Siewiorek, and M. Deisher, "Activity recognition and monitoring using multiple sensors on different body positions," in *Proc. Int. Workshop Wearable Implant. Body Sensor Netw.*, Cambridge, MA, USA, 2006, p. 4 pp.-116
[40] (2015). *Welcome to Lasagne*. [Online]. Available: https://lasagne.readthedocs.io/en/latest/index.html#
[41] M. Abadi. (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. [Online]. Available: https://www.tensorflow.org/
[42] D. Mizell, "Using gravity to estimate accelerometer orientation," in *Proc. 7th IEEE Int. Symp. Wearable Comput.*, White Plains, NY, USA, 2003, pp. 252–253.
[43] A. De Vita, G. D. Licciardo, L. Di Benedetto, D. Pau, E. Plebani, and A. Bosco, "Low-power design of a gravity rotation module for HAR systems based on inertial sensors," in *Proc. IEEE 29th Int. Conf. Appl.-Specific Syst., Archit. Processors (ASAP)*, Milan, Italy, Jul. 2018, pp. 1–4.
[44] A. De Vita, G. Domenico Licciardo, A. Femia, L. Di Benedetto, A. Rubino, and D. Pau, "Embeddable circuit for orientation independent processing in ultra low-power tri-axial inertial sensors," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 67, no. 6, pp. 1124–1128, Jun. 2020, doi: 10.1109/TCSII.2019.2928476.
[45] N. Twomey *et al.*, "A comprehensive study of activity recognition using accelerometers," *Informatics*, vol. 5, no. 2, p. 27, May 2018.
[46] P. Vaidyanathan, S. Mitra, and Y. Neuvo, "A new approach to the realization of low-sensitivity IIR digital filters," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 34, no. 2, pp. 350–361, Apr. 1986.
[47] Z. Wu, Z. Sun, W. Zhang, and Q. Chen, "A novel approach for attitude estimation based on MEMS inertial sensors using nonlinear complementary filters," *IEEE Sensors J.*, vol. 16, no. 10, pp. 3856–3864, May 2016.
[48] J. S. Dai, "Euler–Rodrigues formula variations, quaternion conjugation and intrinsic connections," *Mechanism Mach. Theory*, vol. 92, pp. 144–152, Oct. 2015.
[49] L. E. Emokpae, R. N. Emokpae, and B. Emokpae, "Flex force smart glove prototype for physical therapy rehabilitation," in *Proc. IEEE Biomed. Circuits Syst. Conf. (BioCAS)*, Cleveland, OH, USA, Oct. 2018, pp. 1–4.
[50] R. K. Richards, *Arithmetic Operations in Digital Computers*. Princeton, NJ, USA: D. Van Nostrand Company Inc., 1956.
[51] G. D. Licciardo, C. Cappetta, L. Di Benedetto, A. Rubino, and R. Liguori, "Multiplier-less stream processor for 2D filtering in visual search applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 1, pp. 267–272, Jan. 2018.
[52] G. D. Licciardo, C. Cappetta, L. Di Benedetto, and M. Vigliar, "Weighted partitioning for fast multiplierless multiple-constant convolution circuit," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 64, no. 1, pp. 66–70, Jan. 2017.
[53] STMicroelectronics. (Jan. 2015). *STM32F401xD STM32F401xE*. [Online]. Available: https://www.st.com/resource/en/datasheet/stm32f401re.pdf

Antonio De Vita (Graduate Student Member, IEEE) was born in Avellino, Italy, in 1992. He received the B.Sc. and M.Sc. degrees *(cum laude)* in electronic engineering from the University of Salerno, Fisciano, Italy, in 2015 and 2017, respectively, where he is currently pursuing the Ph.D. degree with the Department of Industrial Engineering. His current research interest includes the design of low-power VLSI systems for digital signal processing in inertial sensors.

Alessandro Russo (Graduate Student Member, IEEE) was born in Avellino, Italy, in 1995. He received the B.Sc. degree *(cum laude)* and the M.Sc. degree *(cum laude)* in electronic engineering from the University of Salerno, Fisciano, Italy, in 2017 and 2019, respectively, where he is currently pursuing the Ph.D. degree with the Department of Industrial Engineering. His current research interest includes integrated low-power hardware accelerators for artificial neural networks.

Danilo Pau (Fellow, IEEE) graduated in electronic engineering from the Politecnico di Milano, in 1992. He joined STMicroelectronics in 1991 working on mpeg2 video memory reduction, then video coding, next embedded graphics, computer vision, and currently on deep learning. He transferred those developments into company products. He is currently the technical director of system research and applications. He has funded and served as a First Chairman for the STMicroelectronics Technical Staff Italian Community. He is also a Fellow Member of IT. Since 2019, he has been serving as an Industry Ambassador Coordinator for the IEEE Region 8 South Europe and the Vice Chair of the Task Force on Intelligent Cyber-Physical Systems within the IEEE CIS. He has contributed with 113 documents the development of *Compact Descriptors for Visual Search* (CDVS), the group that successfully developed ISO-IEC 15938-13 MPEG standard. He was a Funding Chair of the MPEG Ad Hoc Group on Compact Descriptor for Video Analysis (CDVA), formerly Compact Descriptors for Video Search (CDViS). His scientific production consists of 83 articles to date, 78 granted patents, and 23 invited talks/seminars at various universities and conferences.

Luigi Di Benedetto (Member, IEEE) received the B.Sc. and M.Sc. degrees *(cum laude)* in electronic engineering and the Ph.D. degree in solid state electronics from the University of Salerno, Fisciano, Italy, in 2006, 2009, and 2013, respectively. In 2013, he was a Visiting Scientist with the Fraunhofer IISB and Friedrich-Alexander University, Erlangen-Nürnberg, Germany, investigating on blocking behavior of 4H-SiC high-voltage bipolar devices. In 2013, he was a Research Fellow and since 2018, he has been an Assistant Professor of electronic with the Department of Industrial Engineering, University of Salerno. His main research interests include the modeling, simulation, and development of high-power electronic devices based on wide bandgap semiconductorand design of VLSI systems.

Alfredo Rubino (Member, IEEE) received the Laurea degree in physics from the University of Naples Federico II, Naples, Italy, in 1988.

In 1989, he was with the Italian National Agency for New Technologies, Energy, and the Environment, Portici, Italy. Since 2005, he has been an Associate Professor of electronics with the University of Salerno, Salerno, Italy. His current research interests include electronics technology and organic electronics.

Gian Domenico Licciardo (Senior Member, IEEE) received the Electronic Engineering degree from the University of Naples Federico II in 2002 and the Ph.D. degree in information engineering from the University of Salerno, Italy, in 2006. From 2007 to 2018, he was an Assistant Professor of electronic with the University of Salerno, where he joined the Department of Industrial Engineering in 2018 as an Associate Professor. He currently supervises the research activities in the circuital electronic fields, teaches digital and analog electronics to bachelor's and master's students of the electronic engineering courses. He serves as an Coordinator of the IEEE Student Branch of the University of Salerno. He has been involved in many collaborative research projects financed by various instances including private companies and the European Commission. He has published several international journal articles and conference papers about his main research interests which span from the modeling, simulation, and characterization of electron devices to the design of digital VLSI systems for signal processing. He is a member of the Register of Expert Peer Reviewers for Italian Scientific Evaluation (REPRISE). He is an associated editor of several international journals published by IEEE and Springer.